

软件测试—白盒测试 路径测试

路径测试的几种分类

1. 全路径：所有路径的组合情况
2. 独立路径：数学上的基向量，可根据此向量运算出其他任意一条路径

程序图

定义

程序图可看为压缩之后的控制流图，也是一种特殊形式的有向图

绘制控制流图的几个原则

1. 剔除注释语句
2. 剔除变量声明语句
3. 将所有的串行语句压缩为一个节点
4. 所有循环压缩为一次循环，即忽略循环次数

环复杂度

环复杂度（McCabe）是一种程序结构复杂度的度量模型，基于判定节点对程序图封闭环数目造成的影响来衡量程序的复杂程度

计算环复杂度的方法

（1）直观观察法

观察程序图，将二位平面分割为的封闭区域数与开放区域个数，使用此方法需注意，多出口情况的计算

（2）公式计算法

$$V(G) = e - n + 1$$

e表示图中边的数目，n表示图中节点的总数

使用该公式需要满足两个条件：

1. 程序图中不包含孤立的节点
2. 程序图必须是一个强连通图

程序图不为强连通可手动添加辅助路径，将程序图构造为强连通图

（3）判定节点法

利用代码中独立的判定节点数来计算环复杂度，公式如下：

$$V(G) = P + 1$$

基本复杂度

通过将程序图中的结构化设计节点进行不断压缩，最终得到一个无法压缩的程序图，该图的复杂度称为基本复杂度

基本原理

概念

基本原理是通过等价划分来将数据分片，再从每个分片中随意抽取一个数据展开测试，目的是追求测试的完备性与无冗余性

对路径测试的基本原理

将全路径看作一个向量空间，并从全路径集合中抽取的一组线性无关的独立路径作为一组向量基

基路径测试追求的目标

1. 测试的完备性，通过对独立路径测试达到所有路径的条件覆盖
2. 测试的无冗余性，每条路径都是独立的，针对每条路径设计的测试用例之间不存在冗余

独立路径的抽取原则如下：

1. 任意两条独立路径线性无关
2. 所有独立路径的并是整个向量空间

独立路径的抽取

(1) 确定主路径

1. 尽量包含多的循环与判定节点
2. 尽量包含多的语句
3. 该路径尽可能有高的执行效率
4. 尽可能包含复杂的判定表达式

(2) 根据基础路径抽取其他独立路径

基于基础路径，依次在该路径的判定节点执行一个新的分支，构建一条新的独立路径，直到找到足够的独立路径数，即独立路径数=环复杂度

测试用例的设计

1. 根据源代码生成程序图
2. 计算程序图的环复杂度，确定独立路径的集合大小
3. 以最复杂的路径为基础路径，并在此基础上通过覆盖所有判定分支确定其他路径，从而抽取独立路径集合
4. 抽取独立路径时应剔除不可行路径，必要时补充其他重要的路径
5. 根据得到的路径集合对应设计测试用例

不可行路径处理

1. 补充执行概率较高的路径，确保最常执行的项目无缺陷
2. 补充可能包含严重缺陷的路径，例如涉及空间分配
3. 补充涉及复杂算法的路径